Exceptional Model Mining

Duivesteijn, W.

Chapter 5

Deviating Predictive Performance – Classification Model

As a more complex Exceptional Model Mining (EMM) instance, we turn to a classification model. We strive to find subgroups of the dataset for which the performance of a classifier is way off target, or particularly spot-on. In classification models, the output target attribute $y = \ell_m$ is discrete. Generally speaking, the other targets $\ell_1, \ldots, \ell_{m-1}$ can be of any type (binary, nominal, numeric, etc.), though a particular choice of classifier may restrict this. Our EMM framework allows for any classification method, as long as some quality measure can be defined in order to judge the models induced. Although we allow arbitrarily complex methods, such as decision trees, support vector machines or even ensembles of classifiers, we only consider a relatively simple classifier here, for reasons of simplicity and efficiency: we consider the logistic regression model

$$\text{logit}(P(y_i = 1|x_i)) = \ln\left(\frac{P(y_i = 1|x_i)}{P(y_i = 0|x_i)}\right) = \beta_0 + \beta_1 \cdot x_i$$

where $y \in \{0, 1\}$ is a binary class label and $x \in \{\ell_1, \ldots, \ell_{m-1}\}$. The coefficient β_1 tells us something about the effect of x on the probability that y occurs, and hence may be of interest to domain experts. A positive value for β_1 indicates that an increase in x leads to an increase of $P(y = 1|x)$. The strength of influence can be quantified in terms of the change in the odds of $y = 1$ when x increases with, say, one unit.

5.1 Quality Measure φ_{sed}

To judge whether the effect of x is substantially different in a particular subgroup G_D (built from a description D), we fit the model

$$\text{logit}(P(y_i = 1|x_i)) = \beta_0 + \beta_1 \cdot D(i) + \beta_2 \cdot x_i + \beta_3 \cdot (D(i) \cdot x_i) \tag{5.1}$$

where $D(i)$ is shorthand for $D\left(a_1^i, \ldots, a_k^i\right)$ (cf. Section 2.1). Note that

$$\text{logit}(P(y_i = 1|x_i)) = \begin{cases} (\beta_0 + \beta_1) + (\beta_2 + \beta_3) \cdot x_i & \text{if } D(i) = 1 \\ \beta_0 + \beta_2 \cdot x_i & \text{if } D(i) = 0 \end{cases}$$

Hence, we allow both the slope and the intercept to be different for the description and its complement. As a quality measure, we propose to use one minus the p-value of a test on $\beta_3 = 0$ against a two-sided alternative in the model of Equation (5.1). This is a standard test in the literature on logistic regression [84]. We refer to this quality measure as φ_{sed}, an acronym whose meaning is lost in time, but maintained in order to correspond to the acronym in the original paper [71].

5.2 Experiments

5.2.1 Datasets

We demonstrate the classification model on the *Affymetrix* dataset, which was also used in the correlation model experiments. For more details on the dataset, see Section 4.2.1.

5.2.2 Experimental Results

In the logistic regression experiment, we take *NBstatus* as the output $\ell_2 = y$, and *age* (age at diagnosis in days) as the predictor $\ell_1 = x$. The descriptions are created using the gene expression level variables. Hence, the model specification is

$$\begin{gathered} \text{logit}\{\ P(\ y_i = \textit{`relapse or deceased'} \mid x_i\ )\ \} \\ = \\ \beta_0 + \beta_1 \cdot D(i) + \beta_2 \cdot x_i + \beta_3 \cdot (D(i) \cdot x_i) \end{gathered}$$

We find the description

$$D_3 : \mathit{SMPD1} \geq 840 \wedge \mathit{HOXB6} \leq 370.75$$

with a coverage of 33 (52.4%), and quality $\varphi_{\text{sed}}(D_3) = 0.994$. We find a positive coefficient of x for the description, and a slightly negative coefficient for its complement. Within the description, the odds of *NBstatus = 'relapse or deceased'* increase with 44% when the age at diagnosis increases with 100 days, whereas in the complement the odds decrease with 8%. Within the description, an increase in age at diagnosis decreases the probability of survival, whereas within the complement, an increase in age slightly increases the probability of survival. Such reversals of the direction of influence may be of particular interest to the domain expert.

5.3 Alternatives

Another classifier we can consider is the *Decision Table Majority* (DTM) classifier [58, 62], also known as a *simple decision table*. The idea behind this classifier is to compute the relative frequencies of the ℓ_m values for each possible combination of values for $\ell_1, \ldots, \ell_{m-1}$. For combinations that do not appear in the dataset, the relative frequency estimates are based on that of the whole dataset. The predicted ℓ_m value for a new individual is simply the one with the highest probability estimate for the given combination of input values.

Example 1. *As an example of a DTM classifier, consider a hypothetical dataset of* 100 *people applying for a mortgage. The dataset contains two attributes describing the age (divided into three suitable categories) and marital status of the applicant. A third attribute indicates whether the application was successful, and is used as the output. Out of the* 100 *applications,* 61 *were successful. The following decision table lists the estimated probabilities of success for each combination of age and married. The support for each combination is indicated between brackets.*

	married = 'no'	*married = 'yes'*
age = 'low'	0.25 (20)	0.61 (0)
age = 'medium'	0.4 (15)	0.686 (35)
age = 'high'	0.733 (15)	1.0 (15)

As this table shows, the combination married = 'yes' $\wedge$ age = 'low' does not appear in this particular dataset, and hence the probability estimate is based on the complete dataset (0.61*). This classifier predicts a positive outcome in all cases except when married = 'no' and age is either 'low' or 'medium'.*

For this instance of the classification model we discuss two different quality measures. The *Bayesian Dirichlet equivalent uniform* (BDeu) score, which can be used as a measure for the performance of the DTM classifier on G_D, and the *Hellinger distance*, which assigns a value to the distance between the conditional probabilities estimated on G_D and G_D^C.

5.3.1 BDeu Score (φ_{BDeu})

The BDeu score φ_{BDeu} is a measure from Bayesian theory [48] and is used to estimate the performance of a classifier for a description, with a penalty for small contingencies that may lead to overfitting. Note that this measure ignores how the classifier performs on the complement. It merely captures how "predictable" a particular description is.

The BDeu score is defined as

$$\prod_{\ell_1,\ldots,\ell_{m-1}} \frac{\Gamma(\theta/\mathcal{I})}{\Gamma(\theta/\mathcal{I} + \#(\ell_1,\ldots,\ell_{m-1}))} \prod_{\ell_m} \frac{\Gamma(\theta/\mathcal{I}\mathcal{J} + \#(\ell_1,\ldots,\ell_m))}{\Gamma(\theta/\mathcal{I}\mathcal{J})}$$

where Γ denotes the gamma function, $\mathcal{I}$ denotes the number of value combinations of the input variables, $\mathcal{J}$ the number of values of the output variable, and $\#(\ell_1,\ldots,\ell_m)$ denotes the number of records with that value combination. The parameter θ denotes the *equivalent sample size.* Its value can be chosen by the user.

5.3.2 Hellinger (φ_{Hel})

Another possibility is to use the Hellinger distance [115]. It defines the distance between two probability distributions $P(x)$ and $Q(x)$ as follows

$$H(P,Q) = \sum_x \left(\sqrt{P(x)} - \sqrt{Q(x)}\right)^2$$

where the sum is taken over all possible values x. In our case, the distributions of interest are

$$P\left(\ell_m \mid \ell_1, \ldots, \ell_{m-1}\right)$$

for each possible value combination $\ell_1, \ldots, \ell_{m-1}$. The overall distance measure becomes

$$\varphi_{\mathrm{Hel}}(D) = H\left(\hat{P}^{G_D}, \hat{P}^{G_D^C}\right) =$$

$$\sum_{\ell_1,\ldots,\ell_{m-1}} \sum_{\ell_m} \left(\sqrt{\hat{P}^{G_D}(\ell_m|\ell_1,\ldots,\ell_{m-1})} - \sqrt{\hat{P}^{G_D^C}(\ell_m|\ell_1,\ldots,\ell_{m-1})}\right)^2$$

where $\hat{P}^{G_D}$ denotes the probability estimates on G_D. Intuitively, we measure the distance between the conditional distribution of ℓ_m in G_D and G_D^C for each possible combination of input values, and add these distances to obtain an overall distance.

5.3.3 Experimental Results

For the DTM classification experiments on the *Affymetrix* dataset, we have selected three binary attributes. The first two attributes, which serve as input variables of the decision table, are related to genomic alterations that may be observed within the tumor tissues. The attribute *1p_band* (ℓ_1) describes whether the small arm ('p') of the first chromosome has been deleted. The second attribute, *mycn* (ℓ_2), describes whether one specific gene is amplified or not (multiple copies introduced in the genome). Both attributes are known to potentially influence the genesis and prognosis of neuroblastoma. The output attibute for the classification model is *NBstatus* (ℓ_3), which can be either *'no event'* or *'relapse or deceased'*. The following decision table describes the conditional distribution of *NBstatus* given *1p_band* and *mycn* on the whole dataset

1p_band =	*mycn* = *'amplified'*	*mycn* = *'not amplified'*
'deletion'	0.333 (3)	0.667 (3)
'no change'	0.625 (8)	0.204 (49)

In order to find descriptions for which the distribution is significantly different, we run EMM with the Hellinger distance φ_{Hel} as quality measure. As our quality measures for classification do not specifically promote descriptions with larger coverage, we have selected a slightly higher minimum support constraint: *minsup* $= 16$, which corresponds to 25% of the data. The following subgroup of 17 patients (27.0%) was the best found ($\varphi_{\mathrm{Hel}}(D_4) = 3.803$)

$$D_4 : \textit{prognosis} = \textit{`unknown'}$$

1p_band =	*mycn* = *'amplified'*	*mycn* = *'not amplified'*
'deletion'	1.0 (1)	0.833 (6)
'no change'	1.0 (1)	0.333 (9)

Note that for each combination of input values, the probability of *'relapse or deceased'* is increased, which makes sense when the prognosis is uncertain. Note furthermore that the overall dataset does not yield a pure classifier: for every combination of input values, there is still some confusion in the predictions.

In our second alternative classification experiment, we are interested in “predictable” descriptions. Therefore, we run EMM with the φ_{BDeu} measure. All other settings are kept the same. The following subgroup ($|G_5| = 16$ (25.4%), $\varphi_{\mathrm{BDeu}}(D_5) = -1.075$) is based on the expression of the gene *RIF1* (‘RAP1 interacting factor homolog (yeast)’)

$$D_5 : \textit{RIF1} \geq 160.45$$

1p_band =	*mycn* = *'amplified'*	*mycn* = *'not amplified'*
'deletion'	0.0 (0)	0.0 (0)
'no change'	0.0 (0)	0.0 (16)

For this description, the predictiveness is optimal, as all patients turn out to be tumor-free. In fact, the decision table ends up being rather trivial, as all cells indicate the same decision.

5.4 Conclusions

In this chapter, we propose to find descriptions for which a classifier learned from the targets has deviating performance. In theory, this can be done with any classification algorithm, which can be as complex as one desires. In practice, we have developed statistically and probabilistically inspired quality measures for a few relatively simple classification algorithms: logistic regression with merely one predictor, and a multi-predictor Decision Table Majority classifier.

As we have seen in our analysis of description D_3, similarly to some of the findings in the previous chapter, the classification model allows for extensive model inspection. The description merely indicates that a combination of expression level constraints corresponds to deviating behavior. From further model analysis, however, we can learn that the value of one of the predictors has a positive influence on the output value *within* D_3, while the influence is negative *outside of* D_3. This Simpson's paradox is invaluable knowledge for a domain expert.

Specific interest in the resulting subgroups on the dataset domain aside, EMM with this particular model class is potentially extremely interesting within our own field of study, by delivering meta learning information. When data miners are working with, or developing their own, classification algorithms, this instance of EMM can deliver important indications when the algorithm works particularly well, and when it performs not so well. Classification algorithm developers can incorporate this knowledge to improve their algorithms. For classification algorithm users, particularly descriptions such as D_5 found with φ_{BDeu} are potentially interesting. This measure aims to find predictable descriptions. The resulting decision table shows that description D_5 highlights a part of the dataset where predictiveness is optimal: the corresponding subspace of the total search space has essentially been solved. Considering this part of the problem to be solved, we can then focus our attention on classifying the rest of the input space, making the hypothesis space smaller and potentially reducing the computational burden of subsequent classifier runs.